Sniper / Robots - Documentation technique

# A. Contenu

Ce document traite de :

* Création du point d'accès et test de la communication
* Assembler-désassembler le robot
* Recharger les batteries
* Tester le robot
* Précautions avant utilisation

# B. Création du point d'accès

## 1. Matériel

Marque : Conrad - wlan répétiteur pour prise

## 2. Configuration

Pour configurer le point d'accès :

1. brancher le point d'accès à une prise, attendre 1 minute
2. brancher le câble Ethernet
3. ouvrir le navigateur, taper dans la barre d'adresse : 192.168.10.233
4. entrer nom d'utilisateur : admin et mot de passe : admin
5. vous êtes alors sur la page de configuration, utiliser l'assistant de configuration.

Nom du réseau : sniperAProbot.

Mot de passe : razorshark

Le point d'accès permet le transfert de données par Wifi entre les ordinateurs et les robots.

Adresse IPv4: 192.168.10.233

## 3. Communication patch-PD avec point d'accès et module ESP

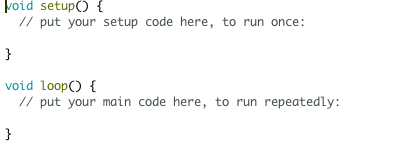
Chaque module ESP a une adresse IP fixe qui correspond au robot sur lequel il est placé (voir marquage sur robot), ainsi le robot 1 a l'IP fixe 192.168.10.11, le robot 2 a l'IP fixe 192.168.10.12 etc.

Le point d'accès route des messages envoyés depuis le patch pure data vers l'adresse correspondante.

Il y a un patch ou un sub-patch par robot.

## 4. Tests module ESP via commandes AP

Procédure de tests :

1. Brancher le module ESP à une carte arduino (avec les modules TEENSY, il est nécessaire de charger des librairies spéciales).
2. Ouvrir un nouveau programme (juste avec setup() et loop () vides), puis le téléverser.
3. Cliquer sur l'icone Moniteur série (au coin de la fenêtre de programmation). 
4. Paramétrer le port série via les menus déroulants au bas de la fenêtre (la vitesse du port x: 115200 bauds et caractères de fin de ligne) .Macintosh HD:Users:lionelstora:Desktop:Capture d’écran 2018-03-13 à 17.10.42.png
5. Taper une commande AT, par exemple : AT, le message de retour doit être OK.
6. Par exemple : AT+CWLAP liste tous les points d'accès visibles par le module.

Pour plus d'infos à propos des commandes AT ESP: <http://domotique.caron.ws/cartes-microcontroleurs/les-commandes-at/>

## 5. Logiciel à installer

Il est nécessaire d'installer 3 logiciels :

* Pure data (PD-extended)
* Arduino version 1.8.4
* Teensyduino

Pure data (PD-extended) est un logiciel Opensource de programmation graphique, il sert d'interface entre le flux de données boursier et les instructions envoyés au robot.

Pour que la communication se fasse correctement, il faut programmer les robots à l'aide du logiciel arduino.

Teensyduino traduit le programme arduino en langage pour la carte Teensy qui contrôle le robot.

Comment installer Teensyduino?

Télécharger l'installateur sur le site : <https://www.pjrc.com/teensy/td_download.html>

Lancer l'installateur (il est nécessaire de connaitre l'emplacement de l'application arduino).

Teensyduino se lance automatiquement sous arduino.

Il suffit de choisir la carte teensy 3.2 dans le menu : Outils > type de cartes > teensy 3.2/3.1

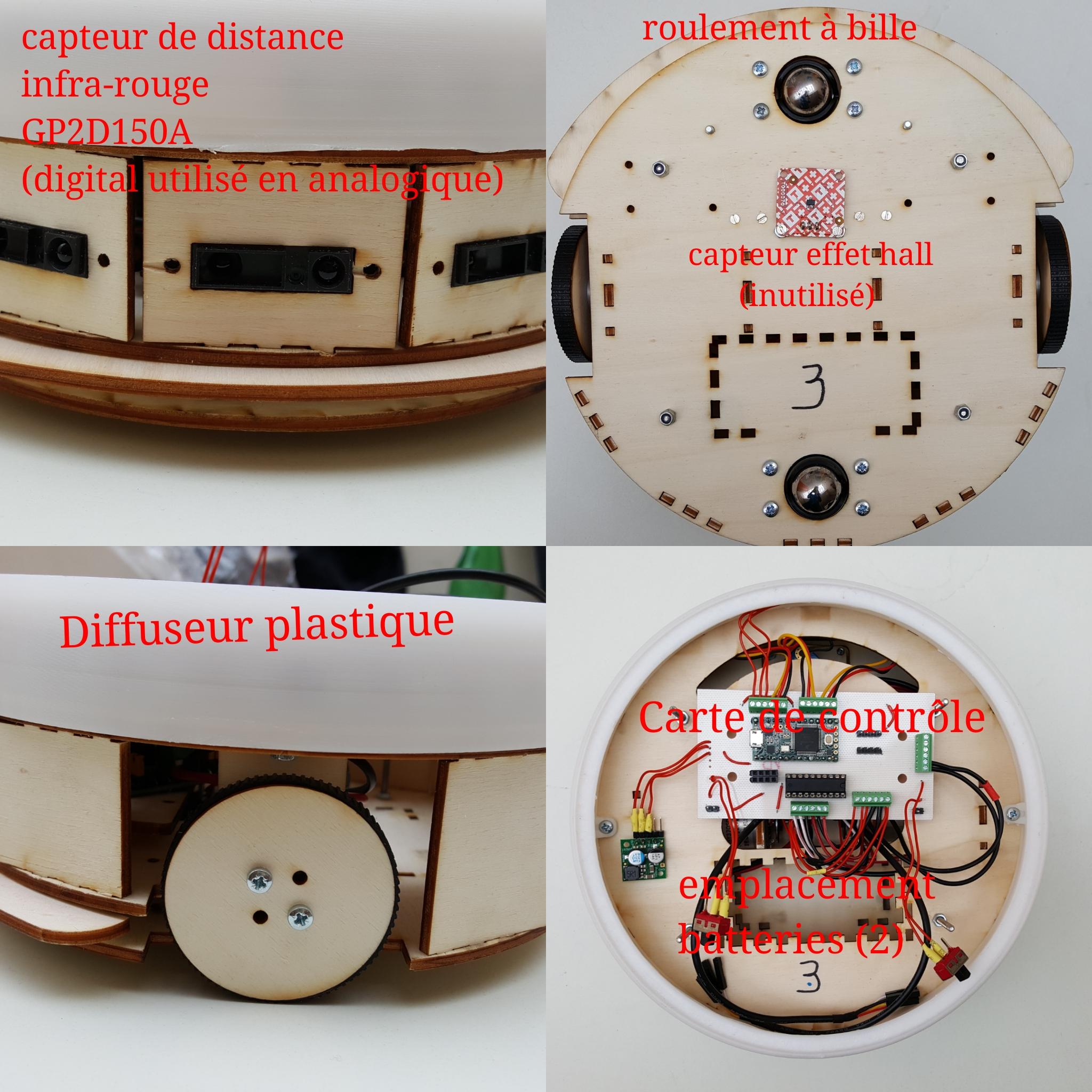
Lors de la compilation sous Arduino, Teensyduino se lance automatiquement.

# B. Montage / Démontage

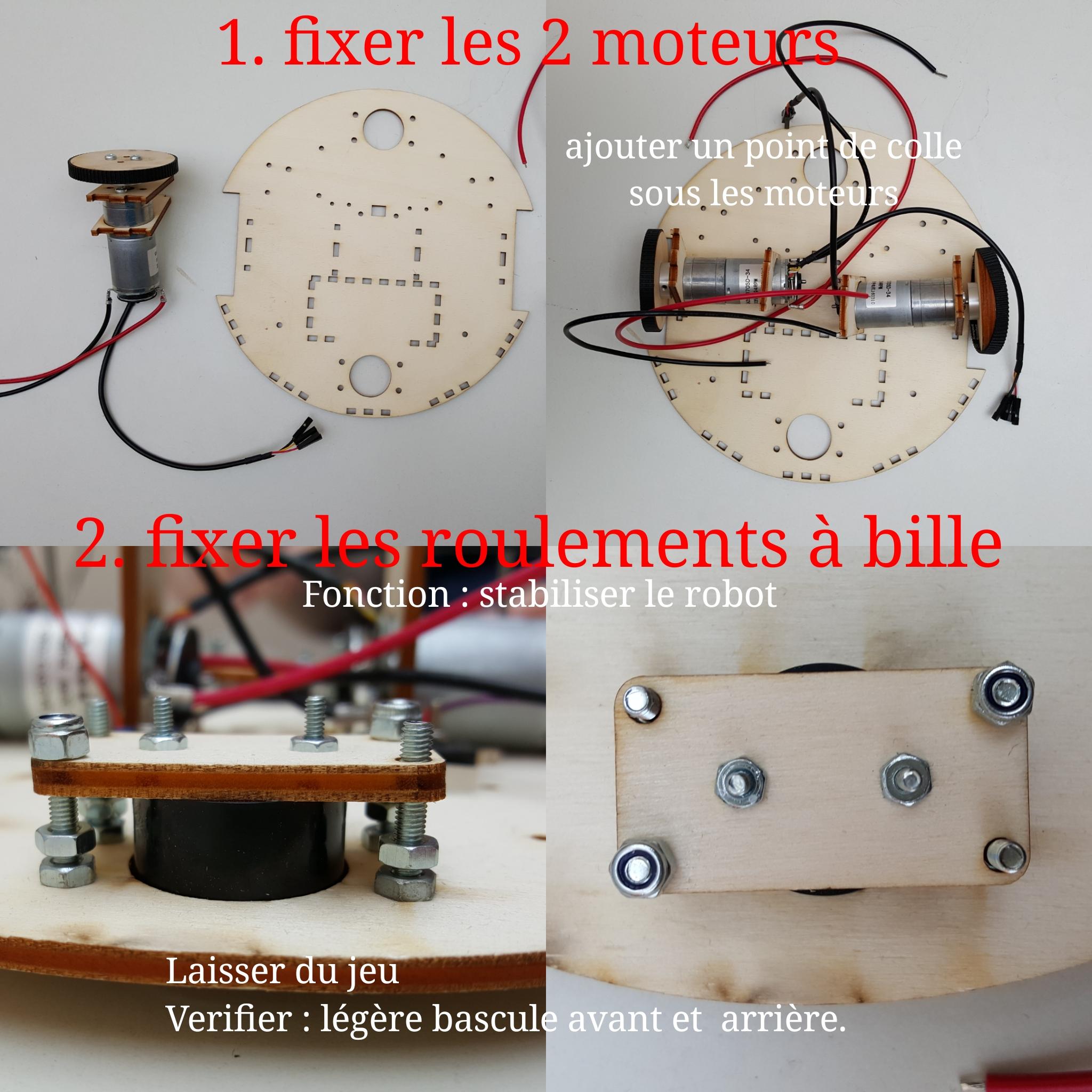
## 1. Vue générale

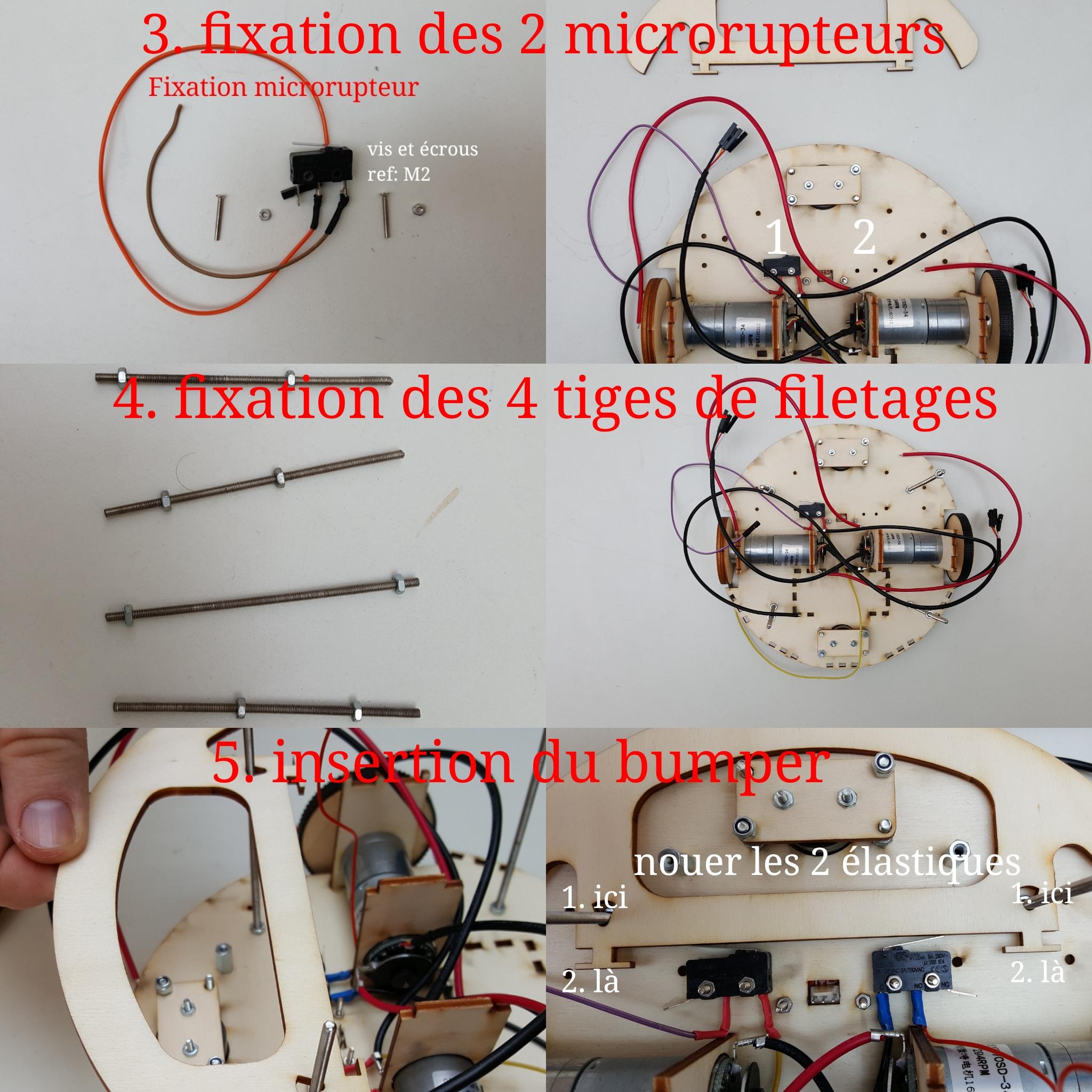
Le robot est constitué de 4 parties :

* le fond sur lequel sont montées les roues, roulement à billes de stabilisation, les moteurs et le pont en H
* un étage intermédiaire qui supporte la carte de contrôle et le module Wifi avec un emplacement d'insertion des batteries
* les capteurs infra-rouges et les panneaux latéraux assujettis entre le fond et l'étage intermédiaire
* le diffuseur et le capot de protection

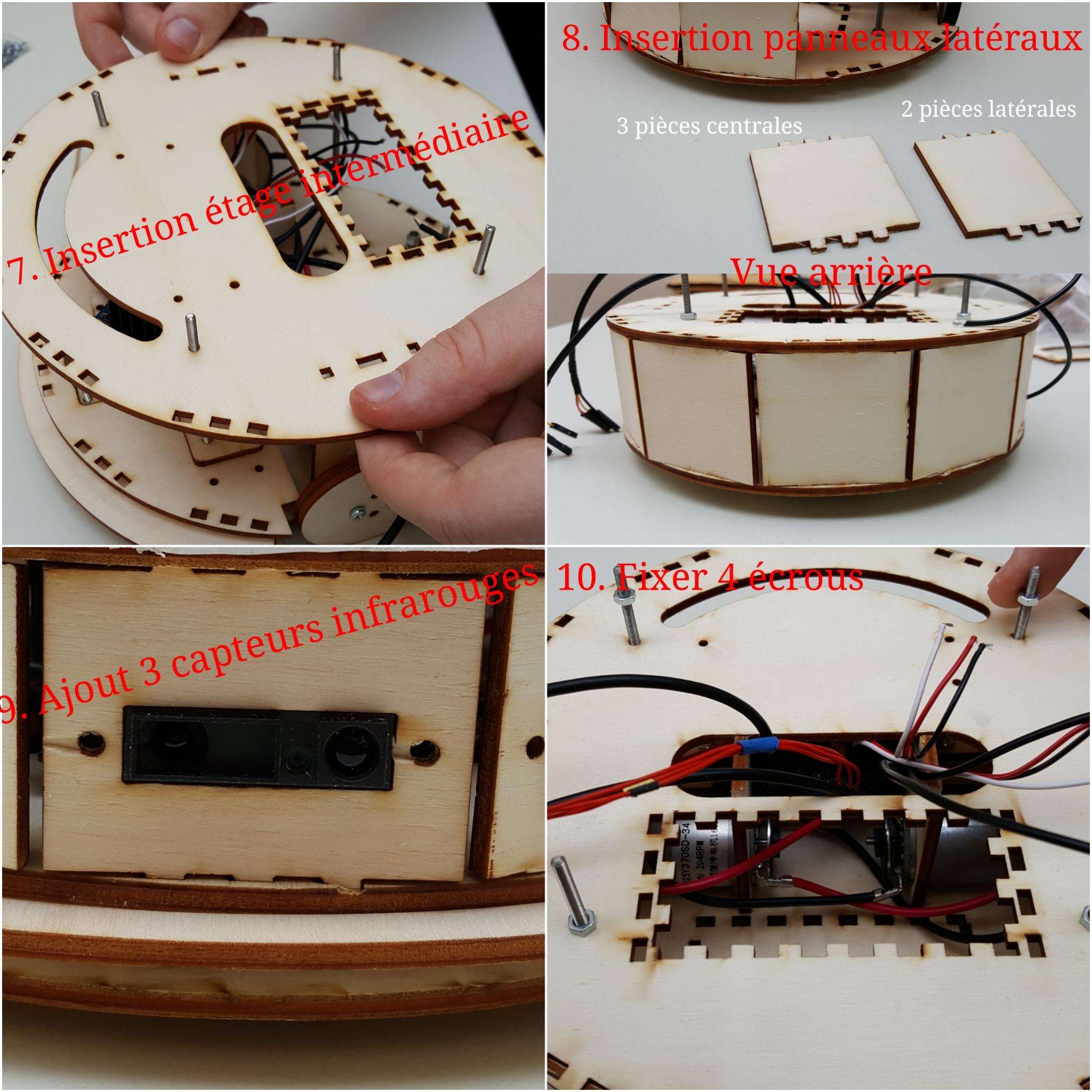


## 2. Assemblage

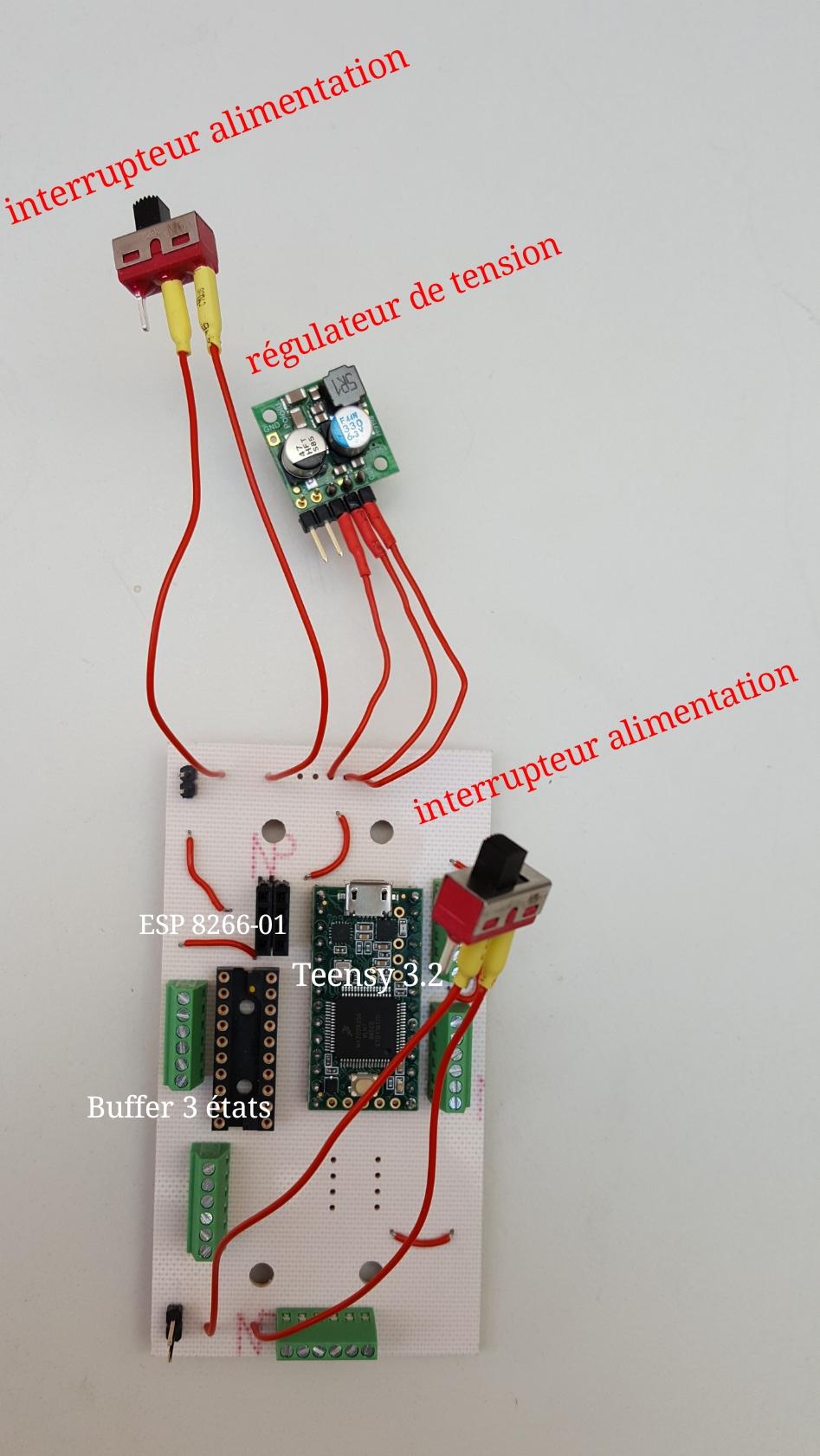


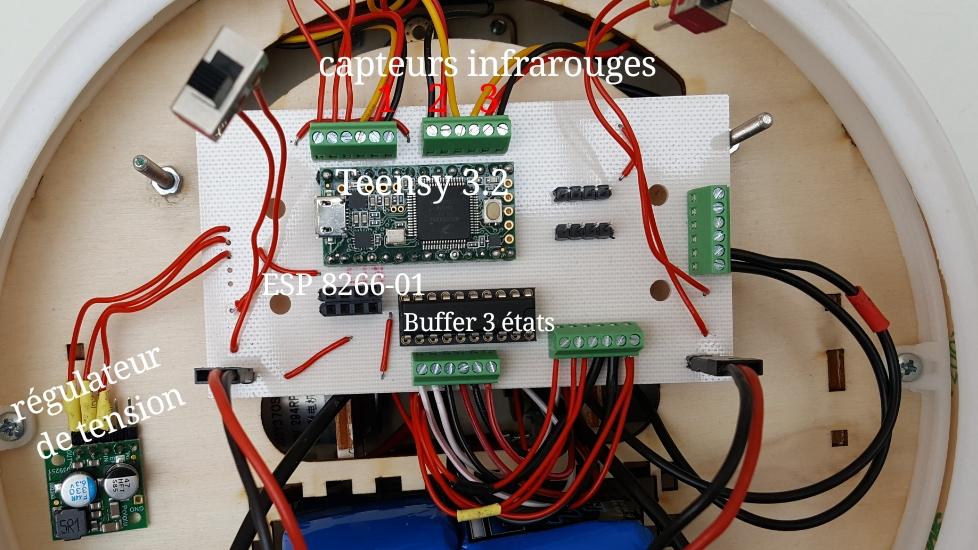


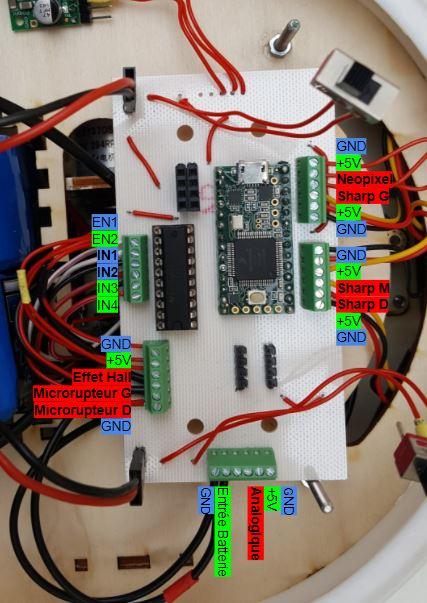
# Macintosh HD:Users:lionelstora:Desktop:PAS:snipper_all:snipper_tech:robot_joshua_20170315:doc_tech_20170314:source_jpg:Manual step3.jpg



# Dernière étape : Câbler la carte de contrôle





  
  
  
  
  
  
(rajouts Théo) **Branchement des codeurs (deux connecteurs centraux). Du haut vers le bas A1 A0 5V GND, puis A3, A2, 5V GND**

**BRANCHER le capteur à effet hall**

**Le connecteur du bas de la photo de gauche à droite (le dessin est faux) GND, Alim moteur, rien, 5V, A5 (analogique), GND**

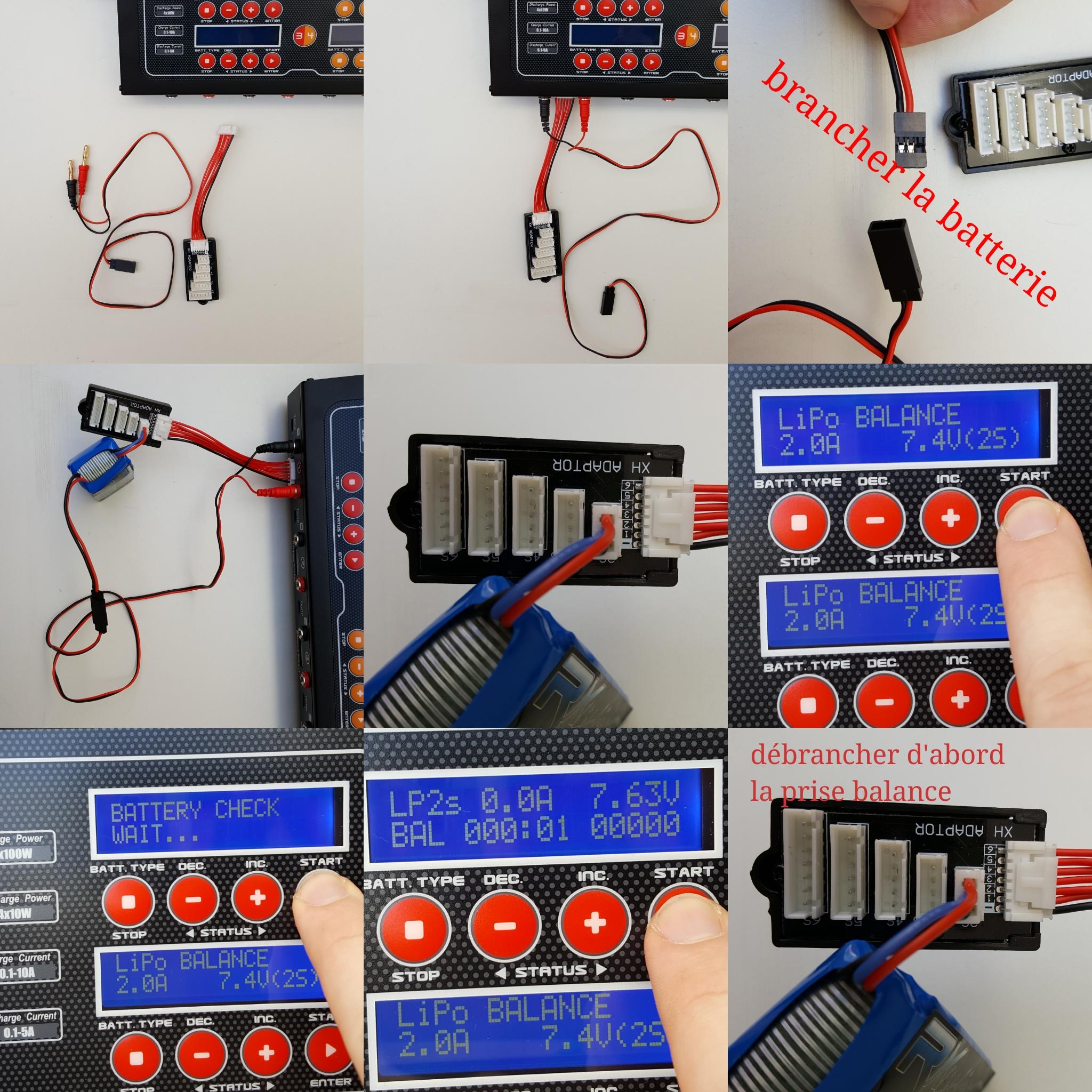
## 3. Câbles et entrées inutilisés en réserve

Pour de futures améliorations, il est prévu :

* 3 câbles regroupés dans une gaine noire (câble noir gauche : encodeur moteur gauche et câble noir droit : encodeur moteur droit, groupe de câble multicouleurs : capteur effet hall)
* sur la droite carte électronique, une entrée analogique est disponible (non-câblée) pour usage ultérieure.
* sur la carte électronique, tourné vers le centre du robot, une entrée analogique est disponible (non-câblée) pour usage ultérieure.

## 4. Installation des batteries

# C. Recharge des batteries



# D. Tests

## 1. Procédure de Test

Procédure :

1. Lancer le logiciel Arduino
2. Brancher la carte Teensy sur l'ordinateur (cable USB - mini USB)
3. Dans l'onglet Outils vérifier les paramètres suivant :

* Type de carte > Teensy 3.2 / 3.1
* USB Type > Serial
* Port > COM13 (Teensy 3.2) Attention nom du port dépend de l'OS utilisé

Si la teensy ne se trouve pas dans la partie Port alors sélectionner Outils > USB Type > Flight Sim Controls, dans cette configuration il n'est pas possible de lire le contenu du port USB mais le programme est transféré de la même manière à la carte teensy.

***Note:*** *A ce jour seul les robots 1, 2 et 3 peuvent communiqué en USB Type > Serial (14.03.18) Ils peuvent donc être utilisés pour du débogage.*

1. Ouvrir un fichier Test avec Fichier > Ouvrir.
2. Compiler puis Téléverser le programme.

## 2. Fichiers de test

Dans le dossier contenant les codes du projet, il y a deux dossiers : un dossier contenant les codes de test de chaque élément et un dossier contenant les codes des robots.

Dans le dossier Test Teensy:

Moteur\_2 >> Fait varier la vitesse des moteurs

Test\_Effet\_Hall >> Lit l'état du capteur

Test\_Teensy >> blynk la led interne de la Teensy

Test\_Teensy\_neopixel >> Allume les LEDs une à une en rouge puis en bleu

Test\_Teensy\_neopixel\_Effet\_Hall >> Allume une led en bleu en cas de présence vu par le capteur, tester avec des aimants (fonctionnelle), avec la bande magnétique (non fonctionnelle)

Test\_Teensy\_neopixel\_microrupteur >> Utilisation de deux leds pour voir le bon fonctionnement des microrupteurs, en rouge et vert quand appuyer en bleu quand rien.

Test\_Teensy\_sharp >> donne les valeurs des capteurs sur le port série (Analogique)

Test\_Teensy\_Sharp\_Digital >> même que précédent mais avec capteur digital

Dans le dossier Test Robot:

Code\_ESP >> code contenu dans les module ESP

Code\_Test\_Teensy\_ESP >> Code permettant de vérifier le bon fonctionnement avec le module ESP en faisant clignoter la led interne de la Teensy

Code\_Test\_Teensy\_ESP\_neopixel >> Prend effet sur la couleur de la bande neopixel (code à utiliser avec pd >> essai\_2)

Esquive\_obstacle >> Code implémenter pour les test d'esquive d'obstacle des robots dans le cas où les capteur son analogique

Esquive\_obstacle\_Digital >> capteur Digital

Test\_complet\_2 >> code utilisant la totalité des fonctions

# E. A faire avant chaque présentation

* Vérifier que les deux roues du robot touche bien le sol, pour cela il suffit d'appuyer sur le dessus du robot au niveau d'une roue puis de l'autre, si il y a balancement alors il faut régler la hauteur des roues billes.
* Vérifier sur chaque roue la présence de la bande noir qui permet une meilleure adhésion.

# F. A Faire

* Tester chaque robot à l'aide des programmes de tests mentionné dans la partie Test.
* Tester l'autonomie des robots en les faisant fonctionner jusqu'à "épuisement".
* Faire des tests sur le problème avec les spots.
* Définir l'utilisation des données boursières.
* Créer le patch PureData qui envoie les différentes données.
* Compléter/Modifier le code principal des robots pour utiliser ses données.
* Tester le code avec un robot, puis trois jusqu'à dix robots dans une zone.